

```
In [ ]: import numpy
import numpy as np
import math
import pandas as pd
from scipy.optimize import minimize
import itertools
from functools import reduce
from gt_design import *
```

Define Ranges

Define the range for the stage loading coefficient, flow coefficient and the reaction. Using the obtained values, calculate the other values.

Use the following data:

1. Reaction (Λ) -> 0.35 - 0.65
2. Flow Coefficient (ϕ) -> 0.6 - 1.0
3. Structural Limit (ΛN^2) -> 10000000 - 22580600 (checking for a wide range)
4. Zweifel for vane -> 0.75 - 0.90
5. Zweifel for rotor -> 0.80 - 0.95

```
In [ ]: range_stage_loading = numpy.linspace(2.5, 4.5, 15)
range_alpha_3 = numpy.linspace(10, 25, 10)
range_mach_exit = numpy.linspace(0.3, 0.55, 10)
range_AN_squared = numpy.linspace(15000000, 22580600, 20)
range_zweifel_vane = numpy.linspace(0.75, 0.90, 4)
range_zweifel_rotor = numpy.linspace(0.80, 0.95, 4)
range_incidence = numpy.linspace(2, 4, 6)
```

```
int main(){
```

```
In [ ]:
```

```
def int_main(range_mach_exit, range_stage_loading, range_alpha_3, range_AN_squared, range_incidence, range_zweifel_vane, range_zweifel_rotor):  
    """  
  
    Design Point Angles:  
    beta_2m -> rotor inlet metal angle  
    beta_3m -> rotor exit metal angle  
  
    Off Design Angles:  
    beta_2 -> rotor inlet flow angle  
    beta_3 -> rotor exit flow angle  
  
    IMP -> FOR DESIGN POINT -> beta_2m = beta_2, beta_3m -> beta_3  
  
    INCIDENCE -> incidence = beta_2 - beta_2m, DEVIATION (for the project) -> beta_3 = beta_3m  
    """>  
  
    data_meanline = []  
    data_root_hub = []  
    data_meanline_losses = []  
    data_off_design = []  
    data_efficiency = []  
    data_scratch = []  
  
    #for i, j, k, l, m, n, o, p in itertools.product(range_mach_exit, range_stage_loading, range_alpha_3, range_degree_reaction, range_AN_squar  
    for i, j, k, m, n, o, p in itertools.product(range_mach_exit, range_stage_loading, range_alpha_3, range_AN_squared, range_incidence, ran  
        incidence = n  
  
        # MEANLINE ANALYSIS  
        T_1, P_1, rho_1, C_1 = aeroturbine.calc_properties(M_1, T_01, P_01)  
        C_a_1 = C_1 * np.cos(np.radians(-10))  
        T_3, P_3, rho_3, C_3 = aeroturbine.calc_properties(i, T_03_cooled, P_03)  
        C_w_1 = numpy.sqrt(C_1**2 - C_a_1**2)  
  
        # Calculate U from the stage loading.  
        U = aeroturbine.calc_U(j)  
        C_a_3, C_w_3, V_3, V_w_3, flow_coefficient_3, beta_3m, a_3, M_3_rel, A_3,P_03_rel = aeroturbine.calc_stage_3(U, C_3, T_3, rho_3,P_03,  
T_02,T_2, P_2, rho_2, A_2, C_a_2, flow_coefficient_2, a_2, V_w_2, beta_2m, V_2, C_w_2, C_2, alpha_2, M_2, M_2_rel,P_02, P_02_rel,l  
if T_02 > 0:  
  
        # STRUCTURAL ANALYSIS  
        A_1 = m_dot_1/(rho_1 * C_a_1)  
        N, omega, r_hub, r_tip, r_meanline, h = aerostructural.calc_structural(m, A_2, U)  
  
        if V_3 > V_2 and C_2 > C_1 and rho_3 < rho_2 < rho_1 and T_02 < T_01 and T_02 > T_03 and T_1 > T_2 and T_2 > T_3 and P_1 > P_2 :  
  
            # BLADE VORTEX ANALYSIS  
            alpha_2_hub, alpha_3_hub, beta_2_hub, beta_3_hub, U_hub, V_2_hub, C_2_hub, M_2_rel_hub, M_2_hub, reaction_hub = aeroturbine  
alpha_2_tip, alpha_3_tip, beta_2_tip, beta_3_tip, U_tip, V_2_tip, C_2_tip, M_2_rel_tip, M_2_tip, M_3_rel_tip = aeroturbine.c  
#increased alpha_2 to 73 from 70  
if 40 < alpha_2 < 75 and U_hub < 335.28 and reaction_hub > 0 and M_2_rel_hub > M_2_rel_tip and 3150 < omega < 4500 and P_02 < 1.01:  
    # ANGLES CHECK  
    if 0 < alpha_2_tip < alpha_2 < alpha_2_hub and 0 < beta_2_tip < beta_2m < beta_2_hub and 0 < alpha_3_tip < k < alpha_3_l  
  
        # OFF DESIGN CALCULATIONS  
        T_3_od,rho_3_od, P_3_od, alpha_3_od, beta_2_od, flow coeff 2 od, incidence 2, v 2 od, C w 3 od,C a 3 od,U mean od,f
```

```

if flow_coeff_3_od > flow_coeff_2_od and C_a_3_od > 0 and 10 <= alpha_3_od <= 40 and P_02_rel_od > P_03_rel_od:

    # VANE AREA AND HEIGHT
    A_vane_mean = (A_1 + A_2)/2
    h_vane_mean = (A_vane_mean * N/60)/U
    r_tip_stator = r_meanline + (h_vane_mean/2)
    r_hub_stator = r_meanline - (h_vane_mean/2)

    # DESIGN POINT LOSSES
    # CALCULATE LOSSES - Stator
    K_p_stator, pitch_chord_ratio_stator, K_accel_stator, stagger_angle_stator, pitch_chord_ratio_stator, pitch_axial_chord_stator = aerodynamic_losses.stator_losses(K_accel_stator, AR_vane, -10, alpha_2)
    K_s_stator = aerodynamic_losses.secondary_losses.calc_K_s(K_accel_stator, AR_vane, -10, alpha_2)
    K_TET_stator, N_stator, c_true_stator, c_a_stator, throat_opening_stator = aerodynamic_losses.trailing_edge_losses.calc_K_TET(N_stator, c_true_stator, c_a_stator, throat_opening_stator)
    K_stator = K_p_stator + K_s_stator + K_TET_stator
    pitch_stator = pitch_chord_ratio_stator*c_true_stator

    # CALCULATE LOSSES - Rotor
    K_p_rotor, pitch_chord_ratio_rotor, K_accel_rotor, stagger_angle_rotor, pitch_chord_ratio_rotor, pitch_axial_chord_rotor = aerodynamic_losses.rotor_losses(K_accel_rotor, AR_rotor, beta_2m, beta_3m)
    K_s_rotor = aerodynamic_losses.secondary_losses.calc_K_s(K_accel_rotor, AR_rotor, beta_2m, beta_3m)
    K_TET_rotor, N_rotor, c_true_rotor, c_a_rotor, throat_opening_rotor = aerodynamic_losses.trailing_edge_losses.calc_K_TET(N_rotor, c_true_rotor, c_a_rotor, throat_opening_rotor)
    K_rotor = K_p_rotor + K_s_rotor + K_TET_rotor
    pitch_rotor = pitch_chord_ratio_rotor*c_true_rotor

    # EFFICIENCY CALCULATIONS DESIGN POINT
    eta_tt = aerodynamic_losses.efficiency_calculations(K_stator, K_rotor, M_2, M_3_rel, C_2, V_3)
    delta_n, eta_final = aerodynamic_losses.efficiency_final(eta_tt, h, beta_3m, r_tip, r_meanline)

    # OFF DESIGN LOSSES
    eta_tt_od, delta_n_od, eta_final_od = aerodynamic_losses.losses_off_design(K_p_rotor, K_s_rotor, K_stator, K_1_rotor, K_2_rotor, K_3_rotor, K_4_rotor, K_5_rotor, K_6_rotor, K_7_rotor, K_8_rotor, K_9_rotor, K_10_rotor, K_11_rotor, K_12_rotor, K_13_rotor, K_14_rotor, K_15_rotor, K_16_rotor, K_17_rotor, K_18_rotor, K_19_rotor, K_20_rotor, K_21_rotor, K_22_rotor, K_23_rotor, K_24_rotor, K_25_rotor, K_26_rotor, K_27_rotor, K_28_rotor, K_29_rotor, K_30_rotor, K_31_rotor, K_32_rotor, K_33_rotor, K_34_rotor, K_35_rotor, K_36_rotor, K_37_rotor, K_38_rotor, K_39_rotor, K_40_rotor, K_41_rotor, K_42_rotor, K_43_rotor, K_44_rotor, K_45_rotor, K_46_rotor, K_47_rotor, K_48_rotor, K_49_rotor, K_50_rotor, K_51_rotor, K_52_rotor, K_53_rotor, K_54_rotor, K_55_rotor, K_56_rotor, K_57_rotor, K_58_rotor, K_59_rotor, K_60_rotor, K_61_rotor, K_62_rotor, K_63_rotor, K_64_rotor, K_65_rotor, K_66_rotor, K_67_rotor, K_68_rotor, K_69_rotor, K_70_rotor, K_71_rotor, K_72_rotor, K_73_rotor, K_74_rotor, K_75_rotor, K_76_rotor, K_77_rotor, K_78_rotor, K_79_rotor, K_80_rotor, K_81_rotor, K_82_rotor, K_83_rotor, K_84_rotor, K_85_rotor, K_86_rotor, K_87_rotor, K_88_rotor, K_89_rotor, K_90_rotor, K_91_rotor, K_92_rotor, K_93_rotor, K_94_rotor, K_95_rotor, K_96_rotor, K_97_rotor, K_98_rotor, K_99_rotor, K_100_rotor, K_101_rotor, K_102_rotor, K_103_rotor, K_104_rotor, K_105_rotor, K_106_rotor, K_107_rotor, K_108_rotor, K_109_rotor, K_110_rotor, K_111_rotor, K_112_rotor, K_113_rotor, K_114_rotor, K_115_rotor, K_116_rotor, K_117_rotor, K_118_rotor, K_119_rotor, K_120_rotor, K_121_rotor, K_122_rotor, K_123_rotor, K_124_rotor, K_125_rotor, K_126_rotor, K_127_rotor, K_128_rotor, K_129_rotor, K_130_rotor, K_131_rotor, K_132_rotor, K_133_rotor, K_134_rotor, K_135_rotor, K_136_rotor, K_137_rotor, K_138_rotor, K_139_rotor, K_140_rotor, K_141_rotor, K_142_rotor, K_143_rotor, K_144_rotor, K_145_rotor, K_146_rotor, K_147_rotor, K_148_rotor, K_149_rotor, K_150_rotor, K_151_rotor, K_152_rotor, K_153_rotor, K_154_rotor, K_155_rotor, K_156_rotor, K_157_rotor, K_158_rotor, K_159_rotor, K_160_rotor, K_161_rotor, K_162_rotor, K_163_rotor, K_164_rotor, K_165_rotor, K_166_rotor, K_167_rotor, K_168_rotor, K_169_rotor, K_170_rotor, K_171_rotor, K_172_rotor, K_173_rotor, K_174_rotor, K_175_rotor, K_176_rotor, K_177_rotor, K_178_rotor, K_179_rotor, K_180_rotor, K_181_rotor, K_182_rotor, K_183_rotor, K_184_rotor, K_185_rotor, K_186_rotor, K_187_rotor, K_188_rotor, K_189_rotor, K_190_rotor, K_191_rotor, K_192_rotor, K_193_rotor, K_194_rotor, K_195_rotor, K_196_rotor, K_197_rotor, K_198_rotor, K_199_rotor, K_200_rotor, K_201_rotor, K_202_rotor, K_203_rotor, K_204_rotor, K_205_rotor, K_206_rotor, K_207_rotor, K_208_rotor, K_209_rotor, K_210_rotor, K_211_rotor, K_212_rotor, K_213_rotor, K_214_rotor, K_215_rotor, K_216_rotor, K_217_rotor, K_218_rotor, K_219_rotor, K_220_rotor, K_221_rotor, K_222_rotor, K_223_rotor, K_224_rotor, K_225_rotor, K_226_rotor, K_227_rotor, K_228_rotor, K_229_rotor, K_230_rotor, K_231_rotor, K_232_rotor, K_233_rotor, K_234_rotor, K_235_rotor, K_236_rotor, K_237_rotor, K_238_rotor, K_239_rotor, K_240_rotor, K_241_rotor, K_242_rotor, K_243_rotor, K_244_rotor, K_245_rotor, K_246_rotor, K_247_rotor, K_248_rotor, K_249_rotor, K_250_rotor, K_251_rotor, K_252_rotor, K_253_rotor, K_254_rotor, K_255_rotor, K_256_rotor, K_257_rotor, K_258_rotor, K_259_rotor, K_260_rotor, K_261_rotor, K_262_rotor, K_263_rotor, K_264_rotor, K_265_rotor, K_266_rotor, K_267_rotor, K_268_rotor, K_269_rotor, K_270_rotor, K_271_rotor, K_272_rotor, K_273_rotor, K_274_rotor, K_275_rotor, K_276_rotor, K_277_rotor, K_278_rotor, K_279_rotor, K_280_rotor, K_281_rotor, K_282_rotor, K_283_rotor, K_284_rotor, K_285_rotor, K_286_rotor, K_287_rotor, K_288_rotor, K_289_rotor, K_290_rotor, K_291_rotor, K_292_rotor, K_293_rotor, K_294_rotor, K_295_rotor, K_296_rotor, K_297_rotor, K_298_rotor, K_299_rotor, K_300_rotor, K_301_rotor, K_302_rotor, K_303_rotor, K_304_rotor, K_305_rotor, K_306_rotor, K_307_rotor, K_308_rotor, K_309_rotor, K_310_rotor, K_311_rotor, K_312_rotor, K_313_rotor, K_314_rotor, K_315_rotor, K_316_rotor, K_317_rotor, K_318_rotor, K_319_rotor, K_320_rotor, K_321_rotor, K_322_rotor, K_323_rotor, K_324_rotor, K_325_rotor, K_326_rotor, K_327_rotor, K_328_rotor, K_329_rotor, K_330_rotor, K_331_rotor, K_332_rotor, K_333_rotor, K_334_rotor, K_335_rotor, K_336_rotor, K_337_rotor, K_338_rotor, K_339_rotor, K_340_rotor, K_341_rotor, K_342_rotor, K_343_rotor, K_344_rotor, K_345_rotor, K_346_rotor, K_347_rotor, K_348_rotor, K_349_rotor, K_350_rotor, K_351_rotor, K_352_rotor, K_353_rotor, K_354_rotor, K_355_rotor, K_356_rotor, K_357_rotor, K_358_rotor, K_359_rotor, K_360_rotor, K_361_rotor, K_362_rotor, K_363_rotor, K_364_rotor, K_365_rotor, K_366_rotor, K_367_rotor, K_368_rotor, K_369_rotor, K_370_rotor, K_371_rotor, K_372_rotor, K_373_rotor, K_374_rotor, K_375_rotor, K_376_rotor, K_377_rotor, K_378_rotor, K_379_rotor, K_380_rotor, K_381_rotor, K_382_rotor, K_383_rotor, K_384_rotor, K_385_rotor, K_386_rotor, K_387_rotor, K_388_rotor, K_389_rotor, K_390_rotor, K_391_rotor, K_392_rotor, K_393_rotor, K_394_rotor, K_395_rotor, K_396_rotor, K_397_rotor, K_398_rotor, K_399_rotor, K_400_rotor, K_401_rotor, K_402_rotor, K_403_rotor, K_404_rotor, K_405_rotor, K_406_rotor, K_407_rotor, K_408_rotor, K_409_rotor, K_410_rotor, K_411_rotor, K_412_rotor, K_413_rotor, K_414_rotor, K_415_rotor, K_416_rotor, K_417_rotor, K_418_rotor, K_419_rotor, K_420_rotor, K_421_rotor, K_422_rotor, K_423_rotor, K_424_rotor, K_425_rotor, K_426_rotor, K_427_rotor, K_428_rotor, K_429_rotor, K_430_rotor, K_431_rotor, K_432_rotor, K_433_rotor, K_434_rotor, K_435_rotor, K_436_rotor, K_437_rotor, K_438_rotor, K_439_rotor, K_440_rotor, K_44
```

run int main

```
In [ ]: data_meanline, data_root_hub, data_meanline_losses, data_off_design, data_efficiency, data_scratch = int_main(range_mach_exit, range_stage_
```

Define Optimization Parameters

```
In [ ]: data_efficiency['delta_eta_optimize_normalized'] = (data_efficiency['delta_eta_optimize'] - data_efficiency['delta_eta_optimize'].min()) /
data_meanline_losses['N_rotor_normalized'] = (data_meanline_losses['N_rotor'] - data_meanline_losses['N_rotor'].min()) / (data_meanline_loss
data_meanline['AN_squared_normalized'] = (data_meanline['AN_squared'] - data_meanline['AN_squared'].min()) / (data_meanline['AN_squared'].m
data_efficiency['eta_opt'] = 1/data_efficiency['eta_final']
data_efficiency['eta_opt_normalized'] = (data_efficiency['eta_opt'] - data_efficiency['eta_opt'].min()) / (data_efficiency['eta_opt'].max())
```

```
In [ ]: # OPTIMIZATION FUNCTION
data_efficiency['func_optimize'] = 0.4 * (data_efficiency['eta_opt_normalized']) + 0.3 * (data_meanline_losses['N_rotor_normalized']) + 0.2
```

Write results to file

```
In [ ]: data_meanline.to_csv('_outputs/output_data_meanline.csv', index=False)
data_root_hub.to_csv('_outputs/output_data_root_hub.csv', index=False)
data_meanline_losses.to_csv('_outputs/output_data_meanline_losses.csv', index=False)
data_off_design.to_csv('_outputs/output_data_off_design.csv', index=False)
data_efficiency.to_csv('_outputs/output_data_efficiency.csv', index=False)
data_scratch.to_csv('_outputs/output_data_scratch.csv', index=False)
```